

WHAT IS CLAIMED IS:

1 1. A method comprising:
2 providing a system including an interface and multiple units of compiled code, the
3 interface including global components and each unit depending on at least one of the global
4 components included in the interface;
5 dividing the interface into levels, each level including a set of one or more of the
6 global components;
7 generating multiple dependency lists;
8 associating a unique one of the multiple dependency lists with each of the levels;
9 associating a unit with a dependency list based on the global components on which
10 the unit depends; and
11 marking only those units associated with a particular dependency list for
12 recompilation based on a change to a particular global component affecting those
13 dependency lists with relationships to a level that includes the changed global component.

1 2. The method of claim 1 wherein the interface includes a definition unit.

1 3. The method of claim 2 further comprising recompiling the unit automatically
2 based on the marking.

1 4. The method of claim 3 wherein recompiling the unit occurs at a subsequent
2 usage.

1 5. The method of claim 4 wherein the subsequent usage is a next usage.

1 6. The method of claim 1 wherein marking only those units associated with a
2 particular dependency list for recompilation based on a change to a particular global
3 component affecting those dependency lists with relationships to a level that includes the
4 changed global component further comprises:
5 determining if a particular property associated with the level has changed; and
6 marking the unit for recompilation only if a particular property has changed.

1 7. The method of claim 1 wherein dividing the interface into levels further
2 comprises assigning an arbitrary number of levels to the interface.

1 8. The method of claim 1 wherein dividing the interface into levels includes
2 assigning a level based on a dependency on all levels of the interface.

1 9. The method of claim 8 further comprising recompiling a client assigned to the
2 level based on a strong dependency on the whole interface after each change to the interface.

1 10. The method of claim 1 wherein dividing the interface into levels further
2 comprises assigning a level based on a dependency on an interface component.

1 11. The method of claim 10 further comprising, recompiling a unit assigned to the
2 level based on a dependency on an interface component after each change to the component.

1 12. The method of claim 11 wherein the change to the component includes a name
2 change.

1 13. The method of claim 11 wherein the change to the component includes a
2 deletion of a component.

1 14. The method of claim 11 wherein the change to the component includes a
2 layout change.

1 15. The method of claim 1 wherein dividing the interface into levels includes
2 assigning a level based
3 on a reference to the interface.

1 16. The method of claim 15 wherein the client depends on the existence of the
2 interface.

1 17. The method of claim 1 further comprising
2 associating indirect clients with a level.

1 18. The method of claim 17 wherein the indirect clients are associated with a
2 lower level than the units.

1 19. The method of claim 1 wherein the dependency list is automatically managed
2 by the system.

1 20. A computer program product, tangibly embodied in an information carrier, for
2 executing instructions on a processor, the computer program product being operable to cause
3 a machine to:

4 provide a system including an interface and multiple units of compiled code, the
5 interface including global components and each unit depending on at least one of the global
6 components included in the interface;

7 divide the interface into levels, each level including a set of one or more of the global
8 components;

9 generate multiple dependency lists;

10 associate a unique one of the multiple dependency lists with each of the levels;

11 associate a unit with a dependency list based on the global components on which the
12 unit depends; and

13 mark only those units associated with a particular dependency list for recompilation
14 based on a change to a particular global component affecting those dependency lists with
15 relationships to a level that includes the changed global component.

1 21. The computer program product of claim 20 further comprising, instructions to
2 cause a machine to recompile the client automatically based on the marking.

1 22. The computer program product of claim 20 wherein the interface includes a
2 definition unit.

1 23. A computer product or article of manufacture configured to:

2 provide a system including an interface and multiple units of compiled code, the
3 interface including global components and each unit depending on at least one of the global
4 components included in the interface;

5 divide the interface into levels, each level including a set of one or more of the global
6 components;
7 generate multiple dependency lists;
8 associate a unique one of the multiple dependency lists with each of the levels;
9 associate a unit with a dependency list based on the global components on which the unit
10 depends; and
11 mark only those units associated with a particular dependency list for recompilation
12 based on a change to a particular global component affecting those dependency lists with
13 relationships to a level that includes the changed global component.

1 24. The computer product or article of manufacture of claim 23 further configured
2 to recompile the unit automatically based on the marking.

1 25. The computer product or article of manufacture of claim 23 wherein the
2 interface includes a definition unit.

1 26. The computer product or article of manufacture of claim 23 further configured to:
2 determine if a property associated with the level has changed, and
3 mark the unit for recompilation only if a property has changed.

1 27. A system comprising:
2 an interface and multiple units of compiled code, the interface including global
3 components and each unit depending on at least one of the global components included in the
4 interface;
5 a means for dividing the interface into levels, each level including a set of one or
6 more of the global components;
7 a means for generating multiple dependency lists;
8 a means for associating a unique one of the multiple dependency lists with each of the
9 levels; a means for associating a unit with a dependency list based on the global components
10 on which the unit depends; and

11 a means for marking only those units associated with a particular dependency list for
12 recompilation based on a change to a particular global component affecting those
13 dependency lists with relationships to a level that includes the changed global component.

1 28. The system of claim 27 further comprising a means for recompiling the unit
2 automatically based on the marking.

1 29. The system of claim 27 wherein the interface includes a definition unit.

1 30. The system of claim 27 further comprising:
2 a means for determining if a property associated with the level has changed, and
3 marking the unit for recompilation only if a property has changed.

1 31. A method comprising the steps of:
2 a step of providing a system including an interface and multiple units of compiled
3 code, the interface including global components and each unit depending on at least one of
4 the global components included in the interface;
5 a step of dividing the interface into levels, each level including a set of one or more of
6 the global components;
7 a step of generating multiple dependency lists;
8 a step of associating a unique one of the multiple dependency lists with each of the
9 levels;
10 a step of associating a unit with a dependency list based on the global components on
11 which the unit depends; and
12 a step of marking only those units associated with a particular dependency list for
13 recompilation based on a change to a particular global component affecting those
14 dependency lists with relationships to a level that includes the changed global component.

1 32. The system of claim 31 further comprising a step of recompiling the unit
2 automatically based on the marking.

1 33. The system of claim 31 further comprising
2 a step of determining if a property associated with the level has changed, and

-
- 3 a step of marking the unit for recompilation only if a property has changed.
-